ELSEVIER
Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia Engineering 15 (2011) 1504 – 1508
Procedia Engineering
www.elsevier.com/locate/procedia

Quick Algorithm for Relative Reduction Based on Decision Information System

Huasheng Zou[a] Changsheng Zhang[b*]

[a] Wenzhou University Oujiang College, Wenzhou, 325035, China
[b] College of Physics and Electronic Information Engineering, Wenzhou University, Wenzhou, 325035, China

Abstract

Computing the core of decision information system and designing efficient relative attribution reduction algorithms are the main research content in rough set theory. After thorough comprehension of the basic concepts of rough set theory, some useful properties based on positive region are found, on this basis, the attribute core is computed by making use of the information of positive region directly. What's more, a computational method of relative reduction is presented. Due to this method has not to calculate discernibility matrix, so as to reduce the cost of time and space, improve the running efficiency of the proposed algorithm. At the same time, theoretical analysis and example analysis illustrate that the efficiency of the proposed algorithm in this paper is improved significantly when compared to the existing algorithms.

Selection and/or peer-review under responsibility of [CEIS 2011]

Keywords: Rough set; Decision information system; Core attribute; Relative reduction

1. Introduction

Since the original exposition of the rough set theory by Pawlak as a new mathematics tool to deal with vagueness and uncertainty of imprecise data[1,2,3]. It has been developed and found applications in the field of decision analysis, fault diagnosis, pattern recognition, data mining and knowledge discovery in database[4,5].The relative reduction is one of the core contents in rough set theory. But the problems of finding all attribute reductions or the minimal attribute reductions in decision system are NP-hard. At present, there is not yet an efficient algorithm for computing optimal attributes or all the attribute reductions. While an efficient relative reduction algorithm is to provide foundation for knowledge discovery in rough set theory, thus finding a quick relative reduction algorithm is still the main research content. Due to many actual application domains there are main decision information systems, denoting as decision table, which is the main research object in rough set theory. Therefore, calculating the core and relative reductions of decision information system is a significant work.

* Corresponding author
E-mail address: zoufan99@163.com

doi:10.1016/j.proeng.2011.08.279

Because in many actual applications, it not guaranteed to get an optimal attribute reduction in decision information system, which is a NP-hard problem. Sometimes it only to compute a sub-optimum attributes reduction. In recent years, some scholars have provided many computing core and relative reduction algorithms[6,7], while more attention has been paid to dicernibility matrix and some improved algorithms based on dicernibility matrix[8,9]. What's more, a method based on information theory is provided in paper[10,11], using information entropy as heuristic knowledge to find a more important attribute. Though intuitively and validly, the disadvantages of these computational methods are that the time complexity and space complexity are too high, which are not adapt to deal with larger data set, so as to restrict the extensive use of rough set theory to a certain degree. After through a comprehensive and profound study of rough set theory, some properties of positive region are found in this paper, at the same time, a method of computing core attribute directly is presented. And a algorithm for relative reduction by using heuristic information based on positive region is put forward. Finally, a practical example is employed to illustrate the efficiency of the new algorithm.

2. Preliminaries

Definition 1. Given an decision information system(a decision table) $S=(U,C,D,V,f)$, where $U=\{x_1,x_2,\cdots,x_n\}$ is the universe of objects, called domain; $A=C\cup D$, where C is a non-empty set of condition attribute, D is a non-empty set of decision attribute, and $C\cap D=\varnothing$; $V=\bigcup_{a\in C\cup D} V_a$, that is, where V_a is the value range of attribute a; f : $U\times C\cup D\to V$ is an information function, which is one information value for each attribute of each object, that is $\forall a\in C\cup D, x\in U,\ f(x,a)\in V_a$ holds. With each attribute subset $B\subseteq(C\cup D)$, there is a binary indiscernibility relation $IND(B)=\{(x,y)\in U\times U\mid \forall a\in B, f(x,a)=f(y,a)\}$, denoted by U/B.

Definition 2. For a decision table $S=(U,C,D,V,f)$, $\forall P\subseteq C\cup D$, denote $U/P=\{P_1,P_2,...,P_m\}$, if there exists $X\subseteq U$, then $P_-(X)=\cup\{P_i\mid P_i\in U/P,\ P_i\subseteq X\}$ is called lower approximation of X with respect to P, $P^-(X)=\cup\{P_i\mid P_i\in U/P,\ P_i\cap X\neq\varnothing\}$ is called upper approximation of X with respect to P.

Definition 3. For a decision table $S=(U,C,D,V,f)$, let $U/D=\{D_1,D_2,\cdots,D_k\}$ be the partition of D with respect to U, $U/P=\{P_1,P_2,\cdots,P_m\}$ be the partition of $P(P\subseteq C)$ with respect to U, $POS_P(D)=\bigcup_{D_i\in U/D} P_-(D_i)$ is called positive region P with respect to D.

Definition 4. For a decision table $S=(U,C,D,V,f)$, $\forall p\in P\subseteq C$, if $POS_P(D)=POS_{P-p}(D)$ then p is not necessary for P with respect to D; Otherwise, then p is necessary for P with respect to D. For $\forall P\subseteq C$, if every element in P with respect to D is necessary, then P with respect to D is independent.

Definition 5. For a decision table $S=(U,C,D,V,f)$, for $\forall P\subseteq C$, if $POS_P(D)=POS_C(D)$, and P with respect to D is independent, , then P is called an attribute reduction of C with respect to D.

Definition 6. For a decision table $S=(U,C,D,V,f)$, denote Red be attribute reduction based on positive region of C with respect to D, then $Core=\bigcap Red$ is called core based on positive region.

Definition 7[7] Let $S=(U,C,D,V,f)$ be a decision table, for $U/C=\{[x'_1]_C,[x'_2]_C,...,[x'_m]_C\}$, then $U' = \{x'_1,x'_2,...,x'_m\}$ and $|[x'_{i_s}]_C/D|=1$ ($s=1,2,...,t$), let $U'_{pos}=\{x'_{i_1},x'_{i_2},...,x'_{i_t}\}$ and $U'_{neg}=U'-U'_{pos}$, it is said that $S'=(U',C,D,V,f)$ is a simplified decision table.

3. Algorithm Description

Computing all reductions or minimal reductions in decision system is a NP-hard problem, because this task is quite difficult to carry out, since it usually needs a very time-consuming search to get the reductions. The needed time is increased rapidly along with the increasing of attribute-scale and domain-scale. Therefore, it is often with the aid of heuristic information to come true a quick calculating for relative reduction. While in decision system the dependency degree of condition attribute with respect to decision attribute is defined as follows:

$$\gamma_C(D)=\sum_{X=U/D}\frac{card(C\_(X))}{card(U)}$$

Where $card(C\_(X))$ denotes the cardinal of object set X, $\gamma_C(D)$ denotes the proportion of domain objects partitioning into the corresponding decision class correctly according to the description of attribute set C. If $Red\subseteq C$, condition attribute $c_i=C-Red$, then the importance degree of attribute c_i is defined as follows:

$$k(c_i,Red,D)=\gamma_{Red\cup\{c_i\}}(D)-\gamma_{Red}(D)$$

According to above definition of the importance degree of attributes , a quick algorithm to compute relative reduction based on relative core is put forward, the algorithm constitutes of two sub-algorithms that is Algorithm 1 and Algorithm 2 respectively.

Algorithm 1: Algorithm for computing relative core based on decision information system

Input：Decision table $S=(U,C,D,V,f)$, where $U=\{x_1,x_2,...,x_n\}$， condition attribute set

$C=\{c_1,c_2,...,c_m\}$， decision attribute set $D=\{d\}$.

Output: Relative core $Core_C(D)$ of decision table S.

Step1：Calculate simplified decision table $S'=(U,C,D,V,f)$；

Step2: Calculate positive region of decision table $POS_C(D)$；

Step3: Let $Core_C(D)=\varnothing$；

Step4: For each condition attribute $c_i\in C$, if $POS'_{C-\{c_i\}}(D)\neq POS'_C(D)$，

Then $Core_C(D)=Core_C(D)\cup\{c_i\}$；

Step5：Output the relative core $Core_C(D)$， then the algorithm terminates；

Because the relative core of decision information system is computed by Algorithm 1, in order to quicken the efficiency of computing relative reduction, on the basis of Algorithm 1, the following algorithm is presented. The algorithm 2 for computing relative reduction based on the computed core attribute set is given as below. And it makes use of positive region as heuristic information for attribute importance degree. What's more, according to the determinate condition of attribute importance degree, the algorithm is to select a more important attribute adding to relative reduction, then satisfying the condition $POS_{Red}(D) = POS_C(D)$, the algorithm terminates, then Red is the relative reduction of decision information system.

Algorithm 2: Quick algorithm for relative reduction based on decision information system

Input: Decision table $S = (U, C, D, V, f)$, $U = \{x_1, x_2, ..., x_n\}$, condition attribute set

$C = \{c_1, c_2, ..., c_m\}$, decision attribute set $D = \{d\}$.

Output: Relative reduction Red of decision table S.

Step1: Calculate simplified decision table $S' = (U, C, D, V, f)$;

Step2: Calculate positive region of simplified decision table $POS'_C(D)$;

Step3: According to Algorithm 1 to compute the relative core of decision table $Core_C(D)$, and let $Red = Core_C(D)$;

Step4: If $Core_C(D) = \varnothing$; then the algorithm turns to Step5, otherwise calculate $POS'_{Red}(D)$, if $POS'_{Red}(D) = POS'_C(D)$, then the algorithm terminates. Output relative reduction Red;

Step5: For each condition attribute $c_i \in C - Red$, calculate $k(c_i, Red, D) = \gamma_{Red \cup \{c_i\}}(D) - \gamma_{Red}(D)$, and select $k(c_i, Red, D) = \max\limits_{c_k \in C - Red} k(c_k, Red, D)$, let $Red = Red \cup c_i$;

Step6: If $POS'_{Red}(D) = POS'_C(D)$, then terminate the algorithm, output relative reduction Red; otherwise the algorithm turns to Step5.

As we known, the relative core of any decision table is alone and it is the intersection of all attribute reductions. In Algorithm 2, we make the relative core as a starting point, and then gradually adding an attribute with a larger classification decision-making ability, then satisfying the relative reduction condition, the algorithm can be obtained the minimal reduction in most cases, at the same time, this reduction is complete.

4. The Example

In this section, in order to illustrate the idea and validity of the new algorithm, we use the following decision table $S=(U,C,D,V,f)$ to illustrate the new algorithm. Where $\{a, b, c, d\}$ are condition attributes, D is a decision attribute, there are all together ten objects. According to the step1 of algorithm 1, we can obtain $U' = \{x_1, x_2, x_5, x_7, x_8\}$, $U'_{pos} = \{x_1, x_2, x_5, x_7\}$, $U'_{neg} = \{x_8\}$, then according to the step4 of the algorithm1, we can get that the core attribute sets are $Core = \{a\}$. On this condition, according to the step3 of the Algorithm 2, let $Red = \{a\}$, since $POS'_{Red}(D) \neq POS'_C(D)$, then for each condition attribute $c_i \in C - Red$, calculate $k(c_i, Red, D) = \gamma_{Red \cup \{c_i\}}(D) - \gamma_{Red}(D)$, and select $k(c, Red, D) = \max\limits_{c \in C - Red} k(c_k, Red, D)$, let $Red = Red \cup \{c\} = \{a, c\}$; so $POS'_{Red}(D) = POS'_C(D)$ then the algorithm terminates. Output relative reduction $Red = \{a, c\}$.

Table 1：Decision table S

U	a	b	c	d	D
x1	2	1	2	1	0
x2	1	2	2	1	1
x3	2	1	2	1	0
x4	1	2	2	1	1
x5	1	1	2	1	1
x6	1	1	2	1	1
x7	1	2	2	1	1
x8	1	2	2	1	1
x9	1	2	2	1	1
x10	1	2	2	1	0

5. Conclusion

To improve the efficiency of the algorithm for relative attribution reduction in rough set theory is a significant work. After thorough comprehension of rough set theory, some useful properties based on positive region are found. The decision properties of core and relative reduction are presented. By making use of positive region as heuristic information to attribution reduction, from the viewpoint of the classification ability of equivalence relation to measure the attribute importance, an efficient method of calculating relative attribution reduction is provided, meanwhile, example analysis illustrates that the proposed algorithm is feasible and efficient.

References

[1] Pawlak Z. Rough set theory and its application to data analysis[J]. *Cybernetics and Systems*, 1998, 9: 661-668.

[2] Pawlak Z, Skowron A. Rudiments of rough sets [J]. *Information Science*, 2007, 117(1):3-37.

[3] Wenxiu Zhang, Guofang Qiu, Weizhi Wu. General theory of attribute reduction in rough set[J]. *Science in China Series E, Technological Sciences*, 2005, 35(12):1304-1313.

[4] Ken Kaneiwa. Sequential pattern mining algorithm using rough set theory[J]. *International Journal of Approximate Reasoning* 2011, 52: 881-893.

[5] Lei Chang, Tengjiao Wang, Dongqing Yang. Efficient algorithms for incremental maintenance of closed sequential patterns in large databases[J]. *Data and Knowledge Engineering*. 2009, 68 (1):68–106.

[6] Shaohui Liu, Qiujian Sheng, Bin Wu. Research on efficient algorithms for rough set methods[J]. *Chinese Journal of Computers*, 2003, 26 (5):524-529.

[7] Zhangyan Xu, Bingru Yang, Wei Song. Comparative study of different attribute reduction based on decision table[J].*Chinese Journal of Electronics,* 2006, 15(4A):953-956.

[8] Chunhsien Chen, Zhiming Rao. A discernibility matrix representation and mapping approach for knowledge acquisition[J]. *Knowledge-Based Systems*, 2008, 21(5): 284-293

[9] Ping Yang. An attribute reduction algorithm based on discernibility matrix[J]. *Information and Control,* 2009, 38(1):70-75.

[10] Guoying Wang. Rough reduction in algebra view and information view[J]. *Journal of Intelligent System*, 2003, 18: 679–688.

[11] Zhangyan Xu, Wei Hou, Bingru Yang. Efficient heuristic attribute reduction algorithm based on information entropy[J]. *Journal of Chinese Computer Systems,* 2009, 30(9):1805-1810.